Projekty – opis i zasady realizacji

Każdy student ma do wyboru realizację jednego z czterech zaproponowanych poniżej projektów. Projekt polega na rozwiązaniu pewnego zagadnienia z wykorzystaniem metod i narzędzi rachunku prawdopodobieństwa. W niektórych projektach należy wykorzystać narzędzia poznane na wykładzie (np. odpowiednie testy statystyczne), w innych trzeba będzie przeprowadzić własny research (np. dotyczący testów serii, nieomawianych na zajęciach).

Niezależnie od tematyki, każdy projekt powinien składać się z następujących czterech etapów, które będą podlegały ocenie przez prowadzącego ćwiczenia (w nawiasie podano maksymalną liczbę punktów, jakie można uzyskać za poszczególne części):

1. **Opis teoretyczny problemu** – należy opisać czego dotyczy projekt, jakie zjawiska będą badane w ramach projektu, sformułować cel projektu, formalnie opisać problem (w języku teorii prawdopodobieństwa, z użyciem zmiennych losowych, rozkładów prawdopodobieństw itp.), przedstawić hipotezy badawcze (max. 3 pkt)
2. **Implementacja rozwiązania** – zestaw kodów źródłowych, skryptów realizujących (implementujących) rozwiązanie, np. klasyfikator, symulator, skrypt uruchamiający określone procedury statystyczne itp. (max. 4 pkt)
3. **Eksperymenty** – opisać zbiór/zbiory danych wykorzystane w eksperymencie, scharakteryzować ten zbiór danych, opisać ewentualne procedury wstępnej obróbki danych, opisać przebieg eksperymentu, przedstawić w odpowiedniej formie wyniki eksperymentów (max. 3 pkt)
4. **Interpretacja wyników** – zinterpretować uzyskane wyniki, przedstawić swoje obserwacje wynikające z eksperymentów, odpowiedzieć na pytania/hipotezy badawcze sformułowane w kroku pierwszym, opisać zagrożenia dla poprawności badań (max. 2 pkt)

Bezwzględny termin oddania projektu to 15 czerwca 2021, godz. 23:59. Rozwiązanie należy przesłać do prowadzącego ćwiczenia mailowo. Na rozwiązanie składa się kod źródłowy oraz raport z badań (zawierający opis pozostałych kroków, tj.: opis problemu, opis eksperymentów, przedstawienie wyników eksperymentów, interpretację wyników).

W terminie 16-26 czerwca odbędą się „obrony" projektów polegające na indywidualnych prezentacjach projektów prowadzącym ćwiczenia. Obrona będzie trwała ok. 15 minut. Celem obrony będzie prezentacja swoich wyników, doszczegółowienie kwestii, które prowadzący uzna za istotne, rozwianie ewentualnych wątpliwości, weryfikacja samodzielności rozwiązania. Jednym z komponentów oceny projektu będzie sposób prezentacji (np. poprawność, formalizm, logiczny układ treści, czytelność, wykorzystanie rysunków i wykresów ułatwiających zrozumienie itp.).

W poniższych opisach projektów podany jest minimalny zestaw wymagań, które należy spełnić, aby uzyskać maksymalną liczbę punktów, jednak należy pamiętać, że na wynik końcowy wpływ będą miały takie czynniki jak w/w: formalizm, poprawność, logiczność wywodów, czytelność, atrakcyjność prezentacji). Student może wykazać się inwencją i wykonać szerszy zakres prac (np. zbadać dodatkowe, postawione przez siebie hipotezy badawcze) – to może mieć wpływ na wynik końcowy.

W przypadku wątpliwości i jakichkolwiek pytań dot. projektów należy kontaktować się z prowadzącymi ćwiczenia (np. mailowo lub na Teamsach).

Każdy student musi wybrać temat projektu i zgłosić go prowadzącemu ćwiczenia do dnia 30 maja.

OPISY PROJEKTÓW

1. Projekt “elf”

Projekt dotyczy analizy zbioru danych dostępnego na Teamsach (plik `logi.zip` w grupie wykładowej). Plik ten zawiera dane o logowaniu się użytkowników na serwer „elf” Instytutu Informatyki i Matematyki Komputerowej. Poszczególne kolumny tego pliku zawierają następujące dane:

- hash MD5 nazwy użytkownika,
- datę logowania na serwer (rok, miesiąc, dzień),
- dzień tygodnia,
- czas rozpoczęcia sesji,
- czas zakończenia sesji,
- całkowity czas sesji.

Należy przeprowadzić statystyczną analizę tych danych. Minimalny zakres prac do wykonania to:

- wstępna kontrola jakości danych (czy występują jakieś braki w danych, czy występują wartości odstające, czy dane zawierają błędne wartości, czy pojawiają się „anomalie” np. dane jednego użytkownika, który uruchamiał bardzo dużo sesji w określonych okresach dnia/tygodnia i czy można tego typu dane usunąć z dalszych analiz)
- graficzna prezentacja danych (np. rozkład całkowitego czasu sesji, rozkład czasu sesji w podziale na dni tygodnia)
- zbadanie, czy rozkład liczby sesji w danym okresie czasu (np. w jednym dniu, w jednym tygodniu) podlega rozkładowi Poissona, a jeśli tak, to dokonać estymacji parametru tego rozkładu
- zbadanie, czy rozkład czasów pomiędzy poszczególnymi logowaniami jest wykładniczy, a jeśli tak, to dokonać estymacji parametru tego rozkładu
- zbadanie, czy pewne wartości (np. liczba, średni czas) w danym okresie czasu są takie same, mniejsze lub większe niż w innym okresie czasu (np. godziny poranne/popołudniowe/wieczorne, dni robocze vs. weekend; okres roku akademickiego vs. wakacje i święta)
- czy w jakichś okresach czasu była zwiększona/zmniejszona aktywność/intensywność logowań (np. sugerujący zbliżający się deadline zadania pod koniec semestru)

Student może weryfikować inne (własne) hipotezy. Podczas badań należy wykorzystać odpowiednie testy statystyczne (przy weryfikacji hipotez) i narzędzia statystyki opisowej.

2. Projekt “klasyfikator”

Projekt polega na stworzeniu klasyfikatora, który na podstawie danych uczących będzie w stanie klasyfikować dane, których wcześniej nie widział. Student może wybrać do implementacji naiwny klasyfikator Bayesowski lub klasyfikator Fishera (liniowa analiza dyskryminacyjna Fishera, LDA) dla klasyfikacji binarnej (w przypadku klasyfikatora Fishera można zaimplementować klasyfikację do wielu klas). W rozwiązaniu nie wolno wykorzystywać gotowych bibliotek/frameworków realizujących działanie klasyfikatorów. Same klasyfikatory należy zaimplementować samodzielnie.

Klasyfikator powinien działać w dwóch trybach: nauki i klasyfikacji. W trybie nauki powinien na wejściu dostawać zbiór uczący z wyróżnioną zmienną oznaczającą poprawną klasę. W trybie klasyfikacji powinien dostawać na wejście zbiór a na wyjściu dla każdego elementu z tego zbioru powinien zwracać przewidzianą klasę.

W ramach eksperymentów należy zbadać skuteczność nauczonego na określonym zbiorze danych klasyfikatora przy pomocy takich miar jak dokładność, precyzja, miara F1, krzywe ROC itp. (zrobić wcześniej research dotyczący metod oceny klasyfikatorów, a także dotyczący metod kroswalidacji)

Do projektu można wykorzystać np. zbiory danych ze strony https://archive.ics.uci.edu/ml/index.php. Po wyborze zbioru (zbiorów) danych (powinny być „interesujące”, nie trywialne) należy przeprowadzić jego (ich) analizę (opisać zmienne (parametry), zbadać ich zależność, korelacje, elementy odstające, zbadać czy w zbiorze są braki itp.), a następnie opracować procedurę uczenia klasyfikatora (np. z zastosowaniem kroswalidacji), przeprowadzić eksperymenty i ocenić jakość klasyfikatora.

3. Projekt “generator liczb losowych”

W ramach tego projektu należy zaimplementować generator G całkowitych liczb pseudo-losowych o rozkładzie równomiernym, oczywiście bez wykorzystania dostępnych funkcji czy bibliotek dla generatorów liczb losowych. Nie wolno również wykorzystywać dostępu do takich źródeł „pseudolosowych” danych jak zegar systemowy. Może to być jeden z prostych generatorów opartych na arytmetyce modularnej. Można zaimplementować więcej niż jeden z takich dostępnych w literaturze generatorów.

Na podstawie generatora G należy następnie stworzyć generator J liczb losowych z rozkładu jednostajnego na przedziale (0, 1), a następnie – na jego podstawie – generatory liczb losowych z rozkładów: Bernoulliego [dwupunktowego] (B), dwumianowego (D), Poissona (P), wykładniczego (W), normalnego (N).

Następnie należy znaleźć w literaturze metody testowania jakości generatorów liczb losowych i wykonać odpowiednie testy dla generatorów G, J, B, D, P, W, N.

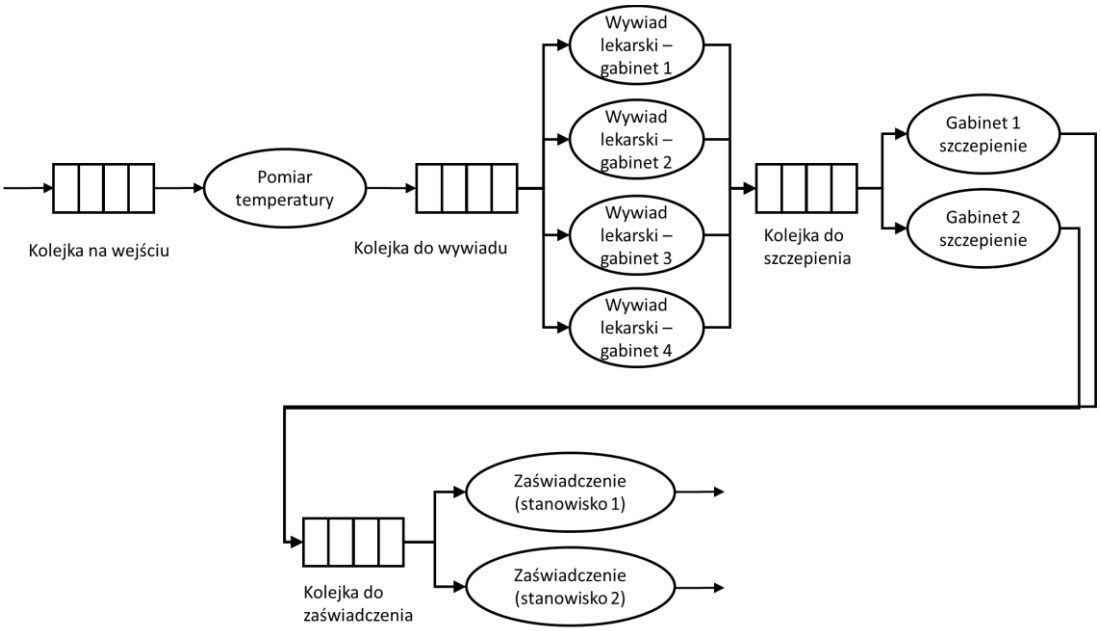
Przydatne materiały:

Test chi-kwadrat zgodności rozkładu
Testy serii
Generator Mersenne Twister
http://home.agh.edu.pl/~chwiej/mn/generatory_16.pdf
https://webhome.phy.duke.edu/~rgb/General/dieharder.php
http://simul.iro.umontreal.ca/testu01/tu01.html

Student może spróbować zaprojektować własny, oryginalny generator G i – przy użyciu testów losowości z powyższych linków – sprawdzić, czy jest on lepszy/gorszy od innych, znanych z literatury prostych generatorów z rozkładu równomiernego wykorzystujących np. arytmetykę modularną.

4. Projekt “Szczepienia w Szpitalu Uniwersyteckim”

Projekt polega na napisaniu symulacji Monte Carlo systemu szczepień w Szpitalu Uniwersyteckim. Należy zaimplementować system kolejkowy dla szczepień działający w SU w Krakowie. System przedstawiony jest schematycznie na poniższym rysunku:



Osoby chcące się zaszczepić przychodzą do kolejki na wejściu z rozkładem Poissona w okresie od godziny 8:00 do 19:00. Pierwsza osoba z kolejki podchodzi do stanowiska pomiaru temperatury w pierwszym momencie, w którym stanowisko to jest wolne. Pomiar temperatury ma rozkład wykładniczy, przy czym jeśli temperatura jest za wysoka, dana osoba jest usuwana z systemu. Ten fakt należy modelować rozkładem Bernoulliego. Po pomiarze temperatury osoby wchodzą do kolejki do wywiadu, z której są po kolei pobierane w momencie, gdy zwolni się stanowisko w jednym z 4 gabinetów lekarskich. Czas trwania wywiadu lekarskiego ma rozkład Gamma. Po wywiadzie osoby wchodzą do wspólnej kolejki do szczepienia, z której są pobierane w momencie zwolnienia się miejsca w jednym z dwóch gabinetów szczepień. Czas szczepienia modelujemy rozkładem chi-kwadrat. Po szczepieniu osoba wchodzi do kolejki po zaświadczenie i pobierana jest gdy zwolni się jedno z dwóch stanowisk do wydawania zaświadczeń. Czas wydania zaświadczenia modelujemy rozkładem jednostajnym. Po otrzymaniu zaświadczenia osoba opuszcza system kolejkowy.

Należy zaimplementować tę symulację i przedstawić jej działanie on-line w formie graficznej (może być uproszczona), dla zadanych parametrów poszczególnych stanowisk obsługi (te parametry powinny być konfigurowalne przez użytkownika). System powinien umożliwiać realizację metody Monte Carlo, tzn. wykonywać odpowiednio dużą liczbę takich symulacji. Należy przeprowadzić kilka takich symulacji estymując: średni czas przebywania w systemie (od wejścia do wyjścia), średnią długość każdej kolejki, średni czas przebywania danej osoby w każdej z kolejek.

Należy też eksperymentalnie zbadać fakt „korkowania się” systemu w zależności od parametrów rozkładów (tzn. zbadać, kiedy długość kolejki na wejściu wydłuża się, a kiedy jest ustabilizowana).